

22/10/2016

السبت

د. ندى

محاضرة 6

## \* Transport Layer

Its output are segments that are sent to network layer

- Transport Layer is on end systems

## # Logical process

يتصل به Layers منظرية

\* وظيفة ال Transport layer وتطبيقها على Logical Comm. بين ال applications أو ال app. processes (يتم به ال processes الموجودة في ال app.)

\* ال Net. Layer يعمل Logical Comm. بين ال end systems  
\* عبارة ال app and trans. Layer عن تطبيق على ال OS منفصلا مع Processes

\* Network Layer → datagram output

\* Transport Layer → segment output

ال Net. Layer فيها IP (internet protocol) كضمانات وخط عليها فتح وتبعها العنوان المطلوب.

\* Network Layer protocol is unreliable ("best-effort")  
does not check Bandwidth, congestion, packet Loss

\* Complexity is in Transport Layer instead of Net. layer  
\* it's up to app. developer to choose trade off between security or speed.

Trans. Layer → TCP (Transmission Control Protocol)  
→ UDP (User datagram Protocol)

\* App developer choose between UDP and TCP

① ~~High throughput, Low delay, no security~~

②

Check 3.4 for comparison UDP vs TCP

TCP is Slower Than UDP because of

① congestion control

② flow control

③ Error checking

TCP (reliable but slow)

UDP (unreliable but fast)

\* IP في ال Net Layer من Reliable و UDP من Unreliable  
Transport Layer في ال Net Layer من Reliability بتزيد بتقيد (complexity)

\* ال Net Layer بتوفر وظائف كثير سريع في الgorithms

\* Transport Layer does Multiplexing/demultiplexing for its segments

Mux: عند اكثر من App قبل واحد من segments فاجمعها  
وابعثها لل Net Layer للتعامل معها

demux: يجمع ال segments من ال Net Layer وبتوزيعها على التطبيقات المختصة

Mux → collect and send

demux → receive and distribute



\* we have sockets between App and trans. layer  
each socket has ID and is application specific

\* when received e.g. email; segment has the  
socket ID so Trans. layer knows which app to  
receive the segment.

\* ~~segment~~ socket ID كامن في buffer، يجمع  
ال segments الخاصة بكل App ويعتبرها واحدة.

\* كل socket له ID مميز يعرفه App وال segments

Check 3.5 for examples

\* Trans layer can do both Mux and demux  
at the same time.

# كل segment له مختلف بين UDP و TCP

headers ← 

↑ SRC Port No.  
↓ dest Port No.

Port No. طول 16 بت، كل Source مع ال dest. 32 bit  
عندي هيزمات، رقم أقدراً تستخدمه ال Application، فال developer لازم يعرف  
ال port اللي شغال عليه

ال Net Layer يجمعها ال IP فقط ومتر بينها بقية الداتا.

\* انه اقل ما تنتقل بين ال layers كل Layer تشوف هيزمات ال headers  
فقط، متر كامل الرسالة، كل رسالة يجمعها ال layer المناظرة  
ليها.

ال SRC Port هو ال Port الی یسكنی علیہ ال Client و ال Server  
على ال application .

Telnet uses UDP , HTTP uses TCP  
check 3.7 for example

UDP uses 8 byte header  
TCP : 20 byte header

- \* UDP is connection-less , unreliable data transfer , no congestion and flow control
- \* it only does checksum (data check) 16 bit by first complement of data.

- یقسم ال data ل 16 بیت

- یجمعها و یجیب ال first complement للجمع (checksum)

- و یأخذ ال checksum یخزنه فی ال header

- ال 16 بیت الثامنة بال checksum هي الی یبنا کد بیوم ص سلامه ال data

- لا یوصله یجمع ال segments مع ال checksum ، لو کل ال نتائج یو طایه (byte of one)  
صیقة ال data صلیقة ، غیر گده ال data صر صلیقة و قهقل .

# Length → to know the segment's size

segments can vary in size

أ - واً مقله فی ال checksum لو صمد تغییر فی مودیم (two bits) صر هتظهر